Linking Sketching and Constraint Checking for Early Conceptual Design

Miles J.C., Cen M. & Taylor M, Cardiff University, Cardiff, UK

Bouchlaghem N.M., Anumba C.J. & Shang H., Loughborough University, Loughborough, UK

Email MilesJC@cf.ac.uk

Abstract

At the start of the conceptual design process, designers start to give tangible form to their thoughts by sketching. This helps with reasoning and communicates ideas to other members of the team. Sketches are gradually worked up into more formal drawings which are then passed to the other stages of the design process. There are however some problems with basing early ideas on sketching. For example, due to their ad-hoc nature, sketches tend only to be diagrammatic representations and so designers cannot be sure that their ideas are feasible and what is being proposed meets the constraints described in the client brief. This can result in designers wasting time working up ideas which prove to be unsuitable. Also the process of constraint checking is complex and time consuming and so designers tend limit their search of possible options and instead choose satisfying rather than good solutions.

This paper describes the INTEGRA project which examines the role of sketching in early conceptual design and how this can be linked to other aspects of the process and particularly automated constraint checking using an IT based approach. The focus for the work is the design of framed buildings. A multi-disciplinary approach has been adopted and the work has been undertaken in close collaboration with practising designers and clients.

1. Introduction

Computer-aided engineering tools such as CAD systems are seldom used in the early stages of the conceptual design due to the ill-defined nature of the information at this stage, and instead sketching plays a key role in early design. Designers use sketching because it offers benefits in terms of fluency, flexibility and speed and at this stage of the design process they are not too concerned about accuracy. Sketching assists with communication, analysis, simulation, and provides an external memory tool thus avoiding cognitive overload (Lipson & Shpitalni, 2000).

Many people have looked at various aspects of the use of sketching tools as conceptual design aids. Landay & Myers (1995) describe the SILK interactive tool, for the conceptual design of user interfaces using freeform strokes. Users sketch a rough prototype of the GUI (graphical user interface) on an electronic pad. The system attempts to recognise these and transform them into a formal interface design in collaboration with the designer.

Zeleznik et.al. (1996) present a gesture-based system, called SKETCH, for constructing three-dimensional views of objects from 2D sketches. In SKETCH, each of the geometric primitive objects such as a cone, cube, cylinder, sphere, pyramid, super-quadric, etc., is associated with a unique gesture. When the user draw curves approximating to these pre-defined gestures, the system automatically creates the corresponding three-dimensional objects and places them in a three-dimensional scene. Another example of a gesture-based system, called *Teddy*, is described in (Igarashi et. al., 1999). In *Teddy*, 3D polygonal surfaces are constructed on the basis of the 2D freeform strokes drawn by the user. The system has been successfully applied to the design of approximate models of stuffed animals and other round objects. However, the system could not create the multiple objects concurrently, combine single objects or design non-round objects such as flowers, houses, etc.

Another type of the system for the construction of the 3D objects from sketching is described in Lipson & Shpitalni (1996) and Lipson & Shpitalni (2000). They propose that humans interpret sketches from experience, thus 3D geometric information about the objects could be obtained from a sketch by the use of learning correlation. The system identifies the configurations of the lines in a sketch and searches the geometric correlation between the various configurations and three-dimensional structures. This information is then decomposed into planar facets and links, together with some other known factors such as the scale factor, material and thickness, from which the system is able to estimate a rough three-dimensional representation of the product.

Fenves et al. (2000) demonstrate a case-based reasoning system, SEED-config, to assist designers with the general conceptual design of a building structure based on the hypothesis that designers usually perform conceptual design using intuition and experience to minimise the amount of time spent on the process. Their system consists of three linked main modules to deal with the architecture, schematic layout, and three-dimensional configuration of the building components. The system provides the user with a set of cases which match the description of the current problem, and designer can select a case which can then be modified as appropriate. The system has not been evaluated by practising designers so its applicability is questionable.

This paper describes the INTEGRA research project part of which examines the role of sketching in early conceptual design and how this can be linked to other aspects of the design process and particularly automated constraint checking using an IT based approach. The focus for the work is the design of buildings. A multi-disciplinary approach has been adopted and the work has been undertaken in close collaboration with practising designers and clients.

In the INTEGRA system, sketching and constraint checking are intimately linked. This is achieved by providing a computer based sketching tool. This uses a plan of the development site as a background for the sketching. As this has a defined scale and is linked to the constraints defined in the client brief, the sketching system understands the size and scope of the problem and so is able to provide immediate feedback on whether the design satisfies the imposed constraints. The constraint checks vary from simple number and area based assessments (e.g. number of car parking spaces, lettable floor area) to far more complex checks (e.g. provision and location of fire escapes). The system is web based so that the designers are able to work remotely while they develop ideas. The sketching system is able to operate in three dimensions so that the floor plan can be developed into elevations and 3D views. These can be linked to visualisation tools which help the designers and the client visualise the impact of the scheme.

2. Conceptual design of buildings

The INTEGRA system has been developed in collaboration with practising designers, contractors and clients. During the early stages of the project a thorough requirements capture process was undertaken with these professionals on aspects of the design process, information handling, constraints, design tools, and communication issues. The findings were used to guide the development of the functionality of the INTEGRA system.

For concurrent engineering, the conceptual design team for a building generally includes a client, an architect, a structural engineer, an M & E engineering and a contractor. The design team primarily focuses on the form, plan layout and structural layout of the building. Sketching is an important tool for the team. The architect and the structural engineer develop and represent design options to other members of the design team using sketches. The design options are represented initially as 2D sketches and those which seem to be the most promising are then transformed into formally drawn plan layouts and 3D views. These options are then checked for constraint compliance. More information about the requirements capture can be found in (Cen, et al., 2003)

3. The Architecture of the INTEGRA system

The INTEGRA system provides an integrated environment for the concurrent conceptual design of a framed commercial building such as a hotel or an office. Apart from the sketching, it has several other tools which cover briefing, risk assessment, rationale capture, filestore handling, and 3D visualization, etc., which facilitate the various aspects of the design process in different ways. Fig.1 illustrates the architecture of the INTEGRA system.

The briefing tool facilitates the initial setting up and the subsequent development of the client brief.. The risk assessment tool elicits and ranks project risks using fuzzy logic and is based on an integration of the project team members' views. This tool is composed of two parts: project server and project client. The project manager starts the risk assessment process on the server side, and team members access to the necessary information from the client system and then submit their inputs to the server via the Internet. The tool then processes them and produces a rank ordered list of risks that the team needs to address (Yang 2001).

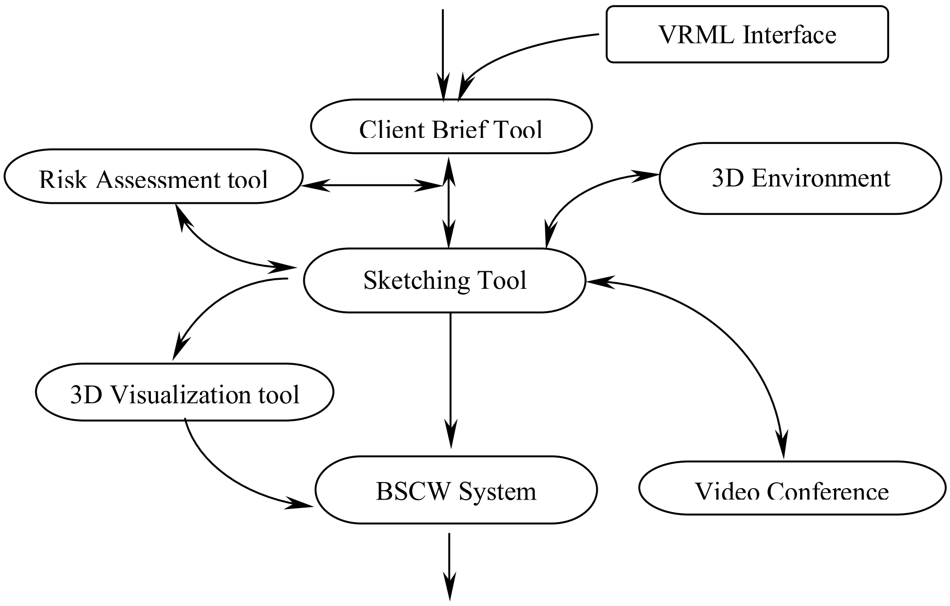

Fig.1 The architecture of the INTEGRA system

The sketching tool is the main subject of this paper and is described below. The 2D sketches from this can be transformed into 3D panoramic views with a 3D visualisation tool based on MGI Photovista software (Shang, et al., 2003).

The communication tools in INTEGRA include Video Conferencing and BSCW. The latter facilitates information transfer between design team members using the notion of a shared workspace, a joint storage facility that may contain various kinds of objects such as documents, tables, graphics, spreadsheets or links to other Web pages. (Shang, et al., 2003). The Video Conferencing tool is part of a novel VR based user interface and allows members to communicate on a one to one or many to many basis (Taylor, et al., 2003).

4. The sketching Tool

The sketching tool contains four main sections these being *Floor plan*, *Elevation View*, *3D Drawing* and *Cost Estimation* and three main functions these being *Free Hand Sketching, Symbol Drawing*, and *Constraint Checking*. In addition there are other functions for editing drawings (Fig. 2). As indicated by the arrows in Fig.2, the *Elevation View* section is linked to the *Floor plan* section, the *Floor plan* section is linked to the *Cost Estimation* section. Fig. 3 illustrates the interface of the sketching tool and *Floor plan* section. The interface for the *Elevation View* is shown in fig.4 The *3D Drawing* interface (not shown) is similar to this but includes features to deal with 3 dimensional sketching.

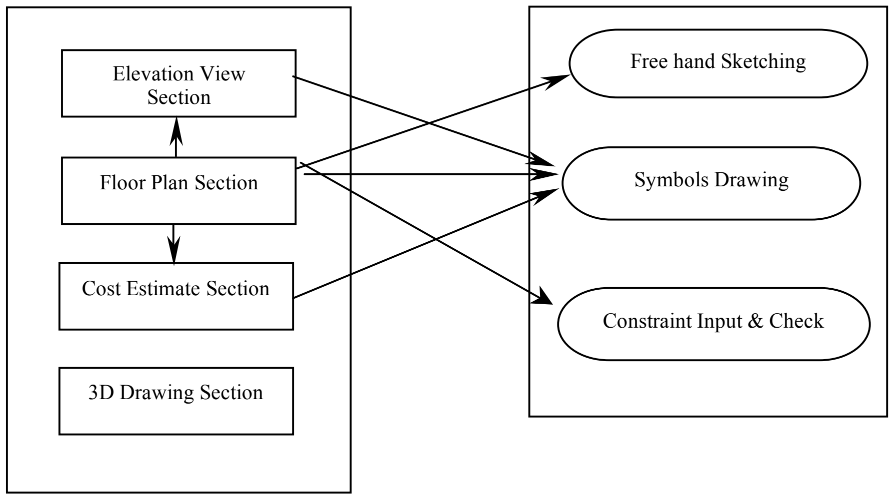

Fig.2 Architecture of the sketching tool

Free Hand Sketching

Free Hand Sketching is available in the *Floor Plan* section. As shown in Fig.3, it includes items for drawing essential shapes, these being *Sketching*, *Line*, *Rectangle*, *Oval,* and *Text*, as well as an *Erase* tool for editing. The *Free Hand Sketching* utility allows the user to draw a line of any form using a mouse. The *line* utility draws straight lines between two points which are defined using the mouse. The *Free Hand Sketching* tool enables users to express and communicate their design ideas and opinions freely.

Symbol Drawing

The *symbol* drawing facility allows designers to draw elements, which can be found commonly in the sketches produced at the conceptual design stage, such as a floor plan sketch. This facility is available for the *Floor plan*, *Elevation View*, *3D Drawing* sections, but the provision in each section varies according to the needs of the different viewpoint.

The *Floor plan* section includes functions which allow the user to draw features such as a *Single Door*, *Double Door, Service Core, Window, Column, Column Group, Car Parkin,, Space,* (used to define features such as standard rooms or column free areas), *Fire Exit* and *Polygon* (see Fig.3). These items are generally found in the floor plan of a typical commercial building such as an office or hotel. In the elevation section, functions such as *Block, Single Door, Double Door* and *Window* are available. The 3D view section also contains *3D Block, 3D Roof, Side Door* and *Side Window*. The advantage in computings terms of having separate symbols for drawing specific features is that they can be given properties and recognised by the computer so it can handle and move them in different ways. This also helps with the constraint checking.

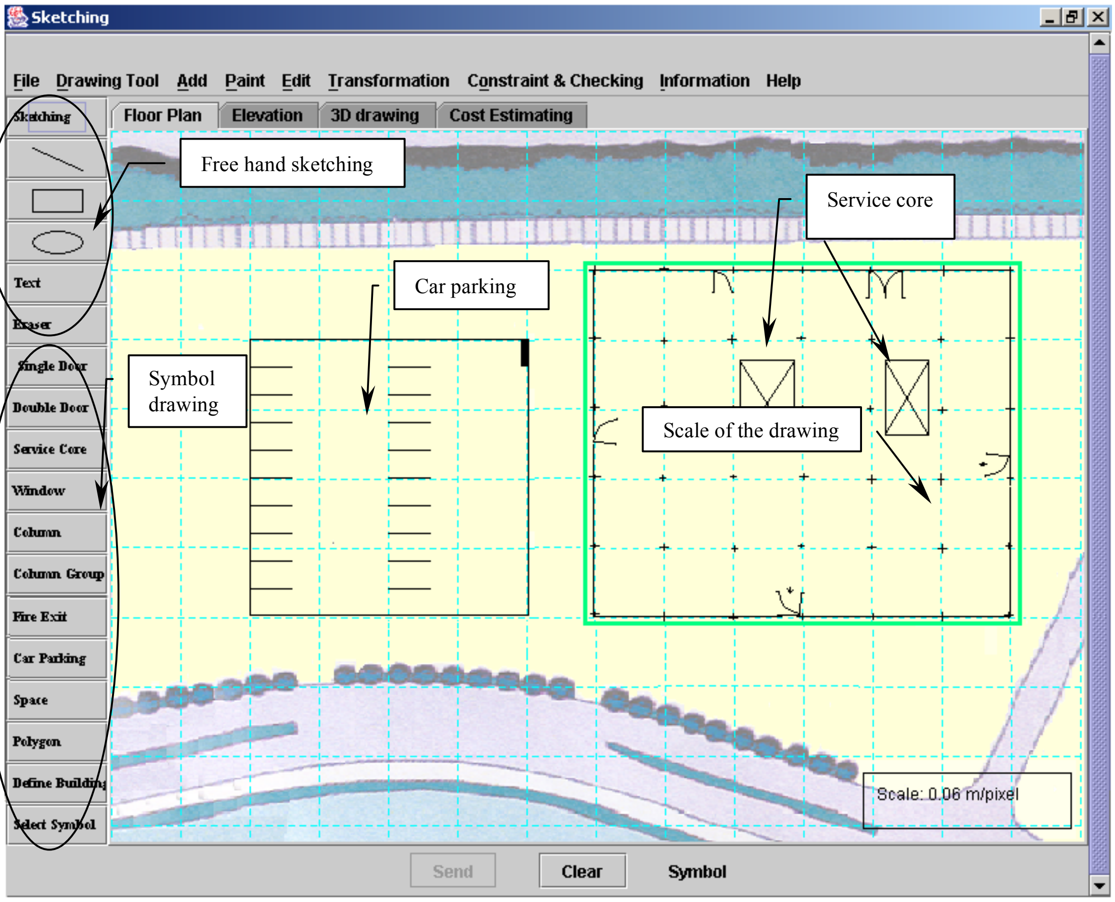

Fig.3 Interface for the sketching tool *Floor Plan* section.

Constraint Input and Check

This function shown in Fig.4 is used for constraint input and checking and includes items from the client brief (e.g. the gross area), fire escape requirements and items dictated by local planning regulations such as limitations on the building height. The constraints for a building design are input by the members of the design team during the design process and are stored in the system design. The client brief also states some of the constraints for the building. Many of the input constraints are linked to the inbuilt constraint checking routines in the software.

The *Client Brief* tool (which is separate from the overall system brief development tool) is where the constraint based information from the client brief is input into the sketching tool and this contains the basic requirements for the building. The *Client Brief* function (see Fig.5) in the sketching tool allows the design team to input requirements relating to gross and net areas, net/gross ratio, and car parking spaces and the project cost. These can either be used just as reference information or in the *constraint checking* utility which is linked to the floor plan sketch.

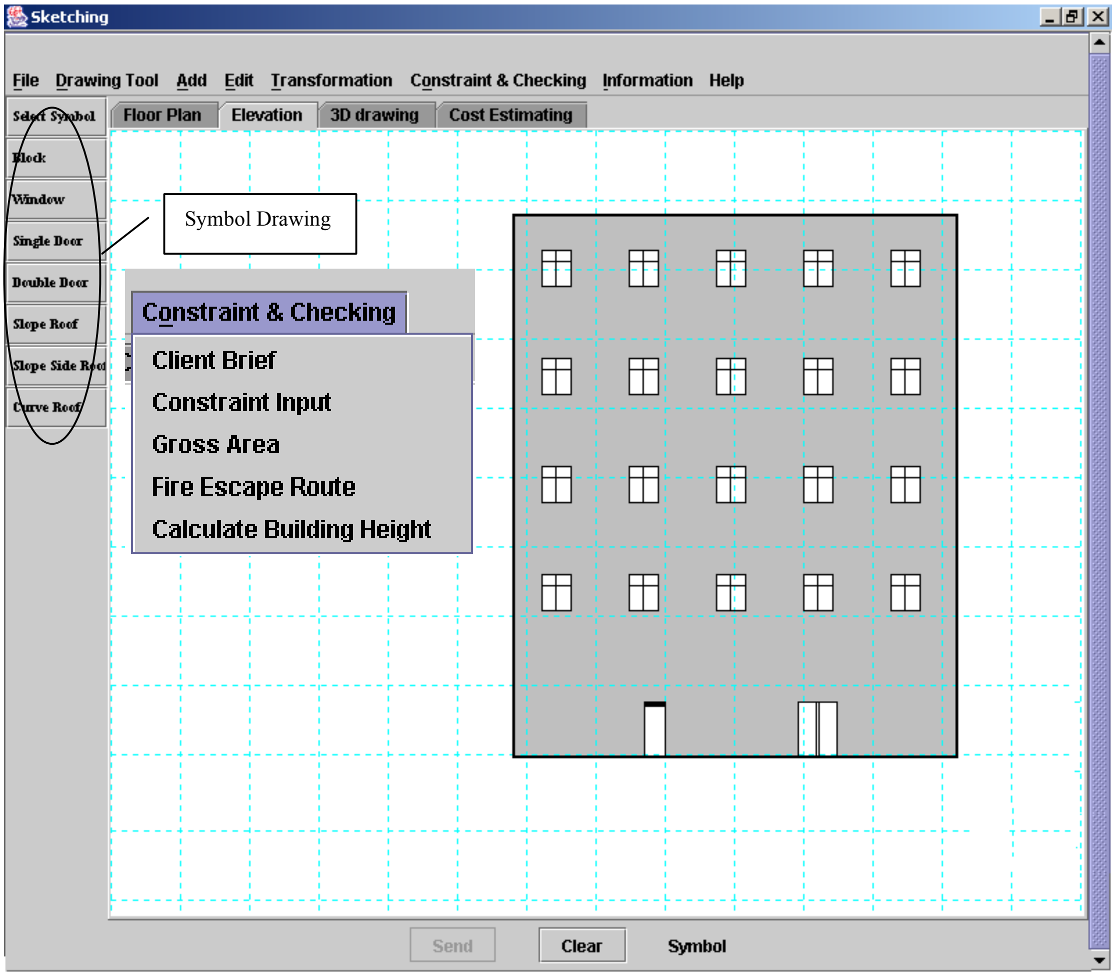

Fig.4 The interface of the *Elevation* section

The purpose of linking the constraint checking with the sketching is so that the designers can easily undertake checks at the very start of the design process and in the subsequent stages. The use by the designers of the *Symbol Drawing* facilities (e.g. door, core column, windows, etc.) enables the system to understand and interpret what has been drawn and give it meaningful properties. This is helped by the fact that the plan view has a facility for inserting a site plan as a background which can be defined as being to a given scale (fig.3). This allows the system to calculate parameters such as the gross area, net area, and number of car parking spaces directly from the sketch. The constraint checking of other parameters, such as project cost, overall building height, and number of stories, requires more information from the users. There are currently only four major areas of constraint checking which have been fully developed (see Fig. 5). These are checking the gross area, assessing compliance with the escape distance rules for fire escapes according to BS5588 part 11, costs and calculating the total height of the building. Further discussion regarding the sketching based constrain checking is contained in the following sections.

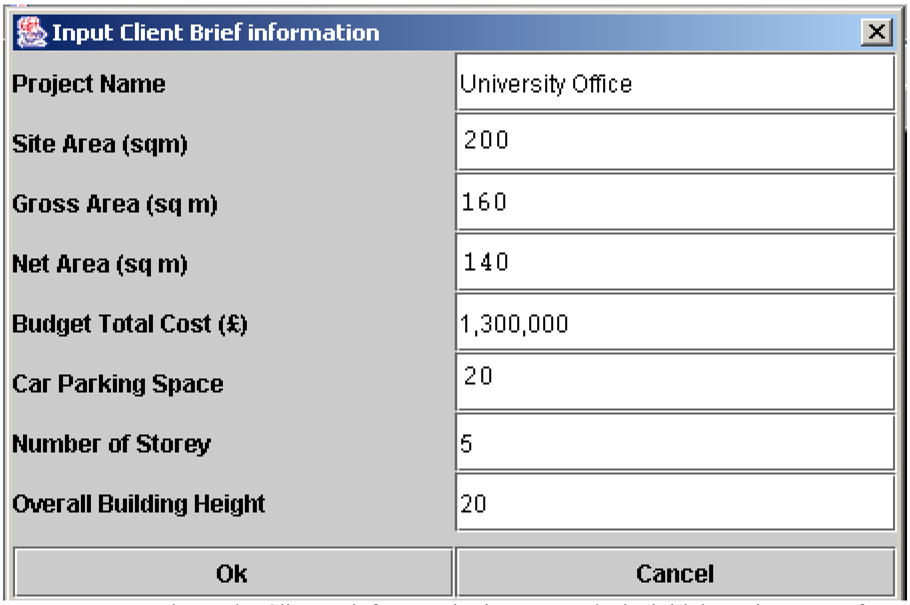

Fig. 5 The Client Brief constraint input records the initial requirements of a project.

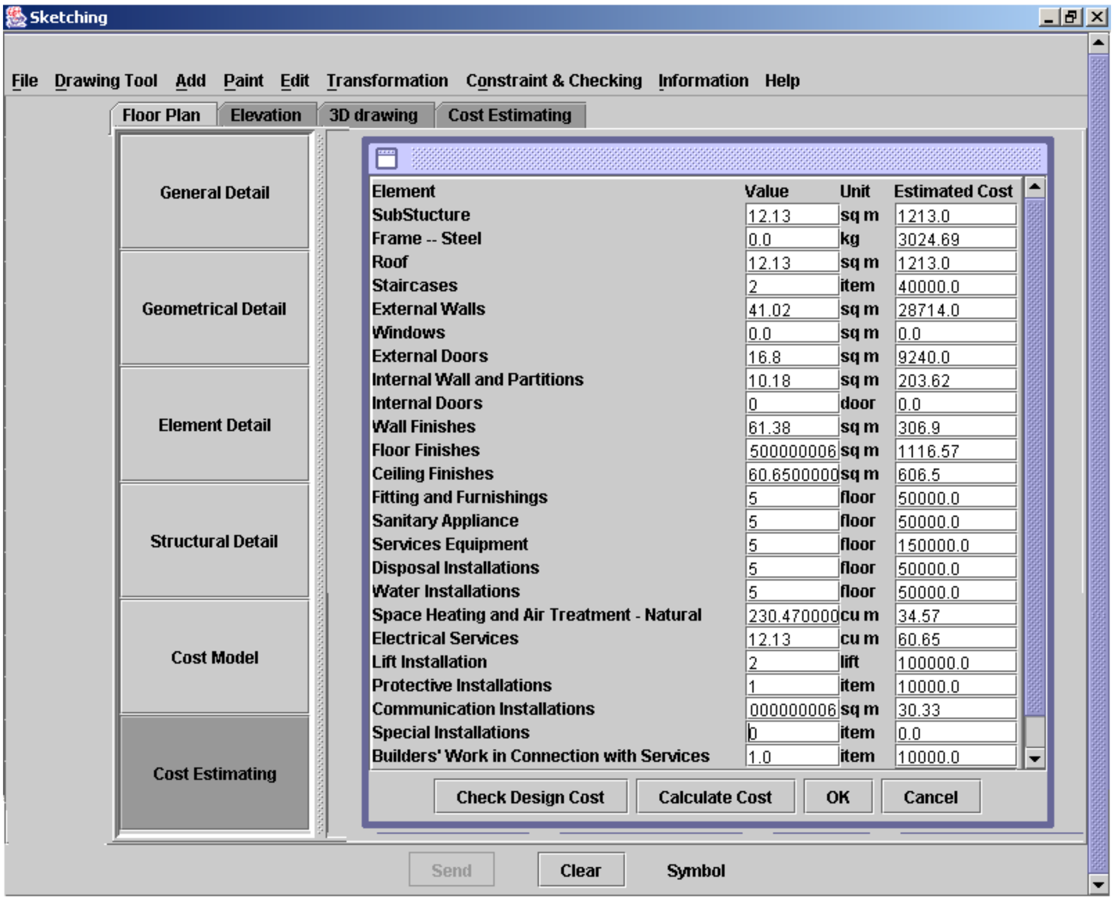

Fig.6 The Cost model

The Cost Model

As discussed above, the sketching tool is linked to an element based cost model (fig.6) for estimating the cost of the building design. The cost model is based on numerous recent observations that 80% of the project cost is contained within the 20% of the most expensive items. These items usually are the same within the class of similar projects. The reference prices for the element costs of office buildings can be found from previous project reports which quantity surveyors or cost managers publish quite regularly in journals, such as *Building* or from specialist cost consultants. The cost of a project can be predicted from the values of these elements.

An example of constraint checking

Fig. 3 illustrates a sketch of the floor plan. At the start of the sketching process, one of the designers has to insert a drawing of the site plan (in the form of a bitmap). This is used as a back ground and also to define the scale of the drawing (currently defined in m/pixel). The designers can then draft their design options using the free hand sketching and symbol drawing tools, although only the latter are linked to the constraint checking facilities.

After completing a sketch of the design option as shown in Fig.3, the users can check the compliance of the design with the constraints. For example, to check the *Gross Area*, the user first has to define the boundary of the building. This is done by activating the *define building* facility (fig.3) and then clicking on the building boundary. The selected boundary is highlighted. The user then pulls down the constraint check menu and selects on the *Gross Area* utility and then clicks on one of the lines that defines the boundary of the building area. The software scans the number of pixels in the chosen shape, calculates the area and a pop-up window appears to provide the user with the location, maximum X and Y dimensions and area of this selected shape. This can then be compared with the area given in the client brief section.

Before the overall building height can be calculated, the users need to input some information regarding the anticipated construction depths for the floors and the voids for such things as the building services and IT (see Fig.7). Using these, the height of the building can then computed as required by activating the relevant utility within the system. Again the results from this can be compared with the requirements in the client brief and the users can decide whether the design option is worth developing further.

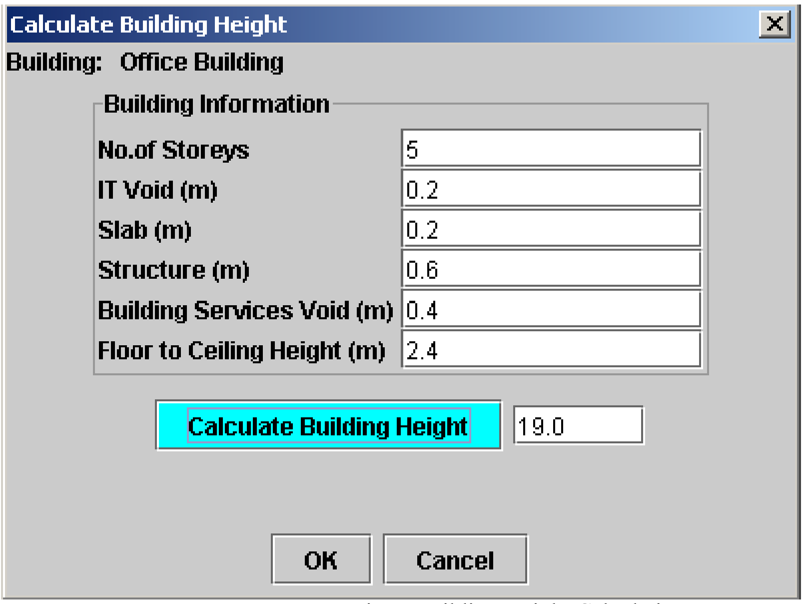

Fig. 7 Building Height Calculation.

Part of a typical result of using the *Fire Escape Route* constraint check is demonstrated in Fig8. For this to give meaningful results, the building floor plan has to have at least one door and for most buildings there have to be at least two. For buildings with more than one floor there has to be two cores and if a core is to be a fire escape then it has to have a door. To activate the *Fire Escape Route* constraint check, the user has to pull down the constraint checking menu (fig.3), choose the fire escape section and and then click on the boundary of the building. Any parts of the design that violate the escape route distance provisions in BS5588, part 11 are then indicated by the system highlighting them in a green shade. The user can examine the design and then amend the escape provision by either moving the cores or providing further cores and doors and then do further constraint checks. In this way it is easily possible to search for good solutions to the fire escape provision problem. In addition to the graphical provision of information the user is also provided with an alphanumeric display (fig.8) which shows the results of the check.

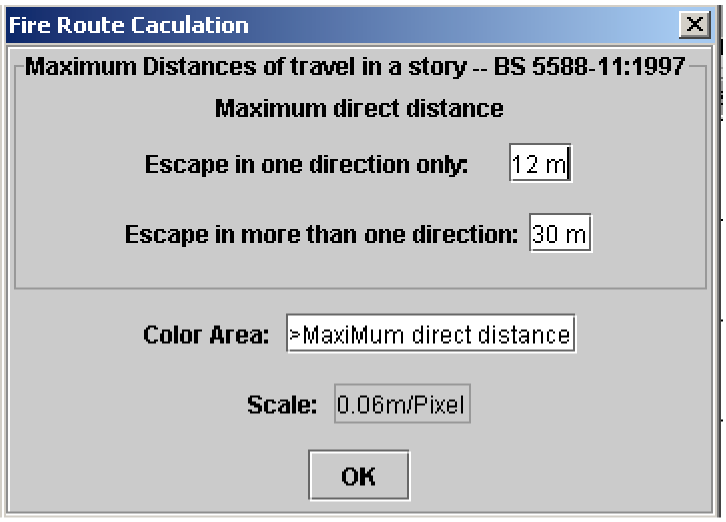

Fig. 8 The result of the *Fire Escape Route* checking.

Fig.6 shows the final results of a cost estimation and part of the overall *cost model* utility. Before this utility can be activated, the users are required to answer some questions on features regarding the design, such as is the building to be open plan or have internal walls, does the building have a steel or concrete frame, etc. The geometrical information about the design, such as dimensions of the building, number and location of the building elements e.g. columns, doors, windows, cores, and car parking, is of course obtained from the sketch. From the sketch, the *cost model* utility works out the other details such as the maximum beam span, the depth of the beam, the areas of walls, etc. and from these data it estimates the cost of the design option. This can then be compared with the project budget given in the *client brief* section.

Elevation and 3D Views

The *plan view* sketching section (fig.3) is linked to the *elevation view* (fig.4) sketching section. After a design option has been drawn using *plan view*, the sketching tool automatically creates the *elevation view* of the option which can be accessed as required by the users. Users can modify the *elevation view*, however, at the moment the changes are not transmitted back to the *plan view*. At present the *elevation view* lacks realism and it is intended to add extra features such as texture and possibly light and shade.

The *3D view* is not linked to the plan view at present and requires the user to add additional information using the 3D sketching functions. At present neither the *elevation view* nor the *3D view* are linked to the constraint checking and there is no plan to do this currently as the principle of linking sketching and constraint checking has been established using the *plan view* and there is plenty of additional work to do on extending the constraint checking abilities for just this.

5. Summary

The architecture and components of the INTEGRA system for the concurrent conceptual design of buildings have been briefly described. The system's various features have been developed following an extensive requirements analysis with practising designers and so the system can be said to represent what their view of an IT support system for conceptual design should contain. When reading this statement, the reader should consider how aware such people are of other developments in providing IT support for conceptual design and whether or not this has influenced the form of the resulting system.

The main focus of this paper is on the *sketching tool* and its linked constraint checks. The *sketching tool* allows users to develop and exchange ideas rapidly. The linked constraint definition and checking tools helps the design team to check the compliance of their design ideas with the client's requirements and other imposed criteria such as local planning laws and national regulations. At present the amount of constraint checking is relatively limited but the system establishes that such checks can easily be linked to and accessed from a sketching tool.

The architecture and main functions of the sketching tool are discussed and examples given of the various constraint checks and how they are linked to the sketching and accessed by the user. Overall the work shows that linking constraints and sketching is feasible and such a process could potentially lead to the fast formation of more formal design descriptions and the early identification of feasible options.

The INTEGRA system is currently being developed further so that the constraint checking will include searches for near optimum solutions using evolutionary computation based techniques. The system has undergone one round of evaluation by practising designers and is about to be subjected to another more through assessment. This process has had a strong influence on the development of the system.

References

Cen, M., Miles, J., Taylor, M., Bouchlaghem, D., Anumba, C.J., D., Yang, H., 2002, Requirement capture for concurrent conceptual design, In: Schnellenbach-Held & Denk (eds.), *Advances in Intelligent Computing in Engineering*, Proc of the 9th EG-ICE Workshop, Darmstadt 2002, pp. 201-211.

Fenves, S.J., Rivard, H. & Gomez N., 2000, SEED-Config: a tool for conceptual structural design in a collaborative building design environment, AI in Engineering 14, pp: 233-247.

Igarashi, T., Matsuoka, S. & Hidehiko, T., 1999, Teddy: a sketching interface for 3D freeform design, In: *Procs SIGGRAPH 1999*, 409-416, ACM, Los Angeles, CA USA.

Landay, J.A. & Myers, A., 1995, Interactive sketching for the early stages of user interface design, In: *Proceedings of CHI'95*, pp.43-50.

Lipson, H. & Shpitalni, M., 2000, conceptual design and analysis by sketching, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 14, pp: 391-401.

Lipson, H. & Shpitalni, M., 1996, Optimization-based reconstruction of a 3D object from a single freehand line drawing, Journal of Computer Aided Design, 28 (8), pp: 651-663.

Shang, H. Anumba, C.J., Bouchlaghem N.M., Miles J.C., Cen,M.& Taylor, M. 2003 An integrated architecture for concurrent conceptual design.

Taylor, M., Miles, J., Cen, M., Bouchlaghem, D., Anumba, C.J., § Shang, H., 2003, A Metaphor Based 3D Interface for a Concurrent Engineering System, the Nashville Conference, USA.

Yang, H.M. (2001) *A Web-based collaborative decision-making system for construction project teams using fuzzy logic*, Ph.D. thesis, Loughborough University.

Zeleznik, R., Herndon, K.P., & Hughes, J.F., 1996, SKETC: An Interface for sketching 3D scenes, In: *Proc SIGGRAPH 1996*, 163-170, ACM , New Orleans, Louisiana.